Create account Log in

Article Talk Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Navigation
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Data item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
العربية
Català
Čeština
Deutsch
Eesti
Español
فارسی
Français
한국어
Italiano
日本語
Norsk bokmål
Polski
Русский
Українська
Tiếng Việt
中文
Edit links

tf–idf

From Wikipedia, the free encyclopedia
(Redirected from Tf*idf)

This article includes a list of references, but **its sources remain unclear because it has insufficient inline citations**. Please help to improve this article by introducing more precise citations. *(July 2012)*

tf–idf, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1]:8 It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

Contents

1 Motivation
2 Definition
3 Justification of idf
4 Example of tf–idf
5 See also
6 References
7 External links and suggested reading

Motivation [edit]

Suppose we have a set of English text documents and wish to determine which document is most relevant to the query "the brown cow". A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document and sum them all together; the number of times a term occurs in a document is called its *term frequency*.

However, because the term "the" is so common, this will tend to incorrectly emphasize documents which happen to use the word "the" more frequently, without giving enough weight to the more meaningful terms "brown" and "cow". The term "the" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less common words "brown" and "cow". Hence an *inverse document frequency* factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Definition [edit]

tf–idf is the product of two statistics, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist. In the case of the **term frequency** tf(t,d), the simplest choice is to use the *raw frequency* of a term in a document, i.e. the number of times that term t occurs in document d. If we denote the raw frequency of t by f(t,d), then the simple tf scheme is tf(t,d) = f(t,d). Other possibilities include[2]:118

- Boolean "frequencies": tf(t,d) = 1 if t occurs in d and 0 otherwise;
- logarithmically scaled frequency: tf(t,d) = log (f(t,d) + 1);
- augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document:

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then

taking the logarithm of that quotient.

$$\operatorname{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

with

- N: total number of documents in the corpus
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\operatorname{tf}(t, d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Mathematically the base of the log function does not matter and constitutes a constant multiplicative factor towards the overall result.

Then tf–idf is calculated as

$$\operatorname{tfidf}(t, d, D) = \operatorname{tf}(t, d) \times \operatorname{idf}(t, D)$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0.

Justification of idf [edit]

Idf was introduced, as "term specificity", by Karen Spärck Jones in a 1972 paper. Although it has worked well as a heuristic, its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it.[3]

Spärck Jones's own explanation didn't propose much theory, aside from a connection to Zipf's law.[3] Attempts have been made to put idf on a probabilistic footing,[4] by estimating the probability that a given document d contains a term t as

$$P(t|d) = \frac{|\{d \in D : t \in d\}|}{N}$$

so that we can define idf as

$$\begin{aligned} \operatorname{idf} &= -\log P(t|d) \\ &= \log \frac{1}{P(t|d)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

This probabilistic interpretation in turn takes the same form as that of self-information. However, applying such information-theoretic notions to problems in information retrieval leads to problems when trying to define the appropriate event spaces for the required probability distributions: not only documents need to be taken into account, but also queries and terms.[3]

Example of tf–idf [edit]

Suppose we have term frequency tables for a collection consisting of only two documents, as listed on the right, then calculation of tf–idf for the term "this" in document 1 is performed as follows.

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

Tf, in its basic form, is just the frequency that we look up in appropriate table. In this case, it's one.

Idf is a bit more involved:

$$\operatorname{idf}(\text{this}, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

The numerator of the fraction is the number of documents, which is two. The number of documents in which "this" appears is also two, giving

$$\operatorname{idf}(\text{this}, D) = \log \frac{2}{2} = 0$$

So tf–idf is zero for this term, and with the basic definition this is true of any term that occurs in all documents.

A slightly more interesting example arises from the word "example", which occurs three times but in only one

document. For this document, tf–idf of "example" is:

$$\mathrm{tf}(\text{example}, d_2) = 3$$
$$\mathrm{idf}(\text{example}, D) = \log \frac{2}{1} \approx 0.3010$$
$$\mathrm{tfidf}(\text{example}, d_2) = \mathrm{tf}(\text{example}, d_2) \times \mathrm{idf}(\text{example}, D) = 3 \log 2 \approx 0.9030$$

(using the base 10 logarithm).

See also [edit]

- Okapi BM25
- Noun phrase
- Word count
- Vector Space Model
- PageRank
- Kullback–Leibler divergence
- Mutual information
- Latent semantic analysis
- Latent semantic indexing
- Latent Dirichlet allocation

References [edit]

1. ^ Rajaraman, A.; Ullman, J. D. (2011). "Data Mining". *Mining of Massive Datasets*. pp. 1–17. doi:10.1017/CBO9781139058452.002 . ISBN 9781139058452. edit
2. ^ Manning, C. D.; Raghavan, P.; Schutze, H. (2008). "Scoring, term weighting, and the vector space model". *Introduction to Information Retrieval*. p. 100. doi:10.1017/CBO9780511809071.007 . ISBN 9780511809071. edit
3. ^ a b c Robertson, Stephen. "Understanding inverse document frequency: On theoretical arguments for IDF". *Journal of Documentation* **60** (5): 503–520.
4. ^ See also Probability estimates in practice in *Introduction to Information Retrieval*.

- Jones KS (1972). "A statistical interpretation of term specificity and its application in retrieval". *Journal of Documentation* **28** (1): 11–21. doi:10.1108/eb026526 .
- Salton G; McGill MJ (1986). *Introduction to modern information retrieval*. McGraw-Hill. ISBN 0-07-054484-0.
- Salton G, Fox EA, Wu H (November 1983). "Extended Boolean information retrieval" . *Communications of the ACM* **26** (11): 1022–1036. doi:10.1145/182.358466 .
- Salton G, Buckley C (1988). "Term-weighting approaches in automatic text retrieval". *Information Processing and Management* **24** (5): 513–523. doi:10.1016/0306-4573(88)90021-0 . Also available at CiteSeerX .
- Wu HC, Luk RWP, Wong KF, Kwok KL (2008). "Interpreting tf–idf term weights as making relevance decisions". *ACM Transactions on Information Systems* **26** (3): 1–37. doi:10.1145/1361684.1361686 .

External links and suggested reading [edit]

- Gensim is a Python library for vector space modeling and includes tf–idf weighting.
- Robust Hyperlinking : An application of tf–idf for stable document addressability.
- A demo of using tf–idf with PHP and Euclidean distance for Classification
- Anatomy of a search engine
- tf–idf and related definitions as used in Lucene
- TfidfTransformer in scikit-learn
- Text to Matrix Generator (TMG) MATLAB toolbox that can be used for various tasks in text mining (TM) specifically i) indexing, ii) retrieval, iii) dimensionality reduction, iv) clustering, v) classification. The indexing step offers the user the ability to apply local and global weighting methods, including tf–idf.
- Pyevolve: A tutorial series explaining the tf-idf calculation .
- TF/IDF with Google n-Grams and POS Tags

Categories: Statistical natural language processing | Ranking functions | Vector space model

This page was last modified on 6 June 2014 at 17:59.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view